ITC 3/53

Information Technology and Control
Vol. 53 / No. 3/ 2024
pp. 798-812
DOI 10.5755/j01.itc.53.3.34856

Effect Analysis of Malicious Flow Classification Model Based on Representation Learning on Network Flow Anomaly Detection

Received 2023/08/14 | Accepted after revision 2024/04/23

HOW TO CITE: Hu, Y., Duan, X., Chen, Y., Zhao, Z. (2024). Effect Analysis of Malicious Flow Classification Model Based on Representation Learning on Network Flow Anomaly Detection. *Information Technology and Control*, 53(3), 798-812. https://doi.org/10.5755/j01.itc.53.3.34856

Effect Analysis of Malicious Flow Classification Model Based on Representation Learning on Network Flow Anomaly Detection

Yan Hu

College of Intelligent Transportation, Hunan Communication Polytechnic, Changsha, 410132, China

Xiaole Duan

Teaching Department of Public Courses, Hunan Communication Polytechnic, Changsha, 410132, China

Yuan Chen, Zhu Zhao

College of Intelligent Transportation, Hunan Communication Polytechnic, Changsha, 410132, China

Corresponding author: Yan_Hu2023@outlook.com

Network traffic anomaly detection, as a key link of network security, has been paid more and more attention in recent years. Aiming at abnormal flow caused by improper network usage, this paper proposes a network flow anomaly detection model using representation learning. In this model, the study treats raw flow data as images directly through representation learning, and then classifies malicious flow by performing image classification tasks. The study is tested using the USTC-TFC2016 dataset. The experimental results show that the model exhibits excellent classification accuracy of 0.9990 both in the characterization of flow sessions and total flow, and PR and F1 values are all above 0.9907. In addition, the classification accuracy of the three classifiers for flow data is more than 98%, and the classification accuracy of normal flow and malicious flow is 100%. The experimental results show that the proposed method meets the needs of practical applications and has excellent classification performance. This provides a new research angle and direction for network flow anomaly detection.

KEYWORDS: RL; Anomaly detection; Flow classification; Convolution neural network.

1. Introduction

With the continuous development of information technology, the Internet has an important relationship with people's interests and has become an indispensable infrastructure for people. At the same time, the rise of malicious flow has brought huge impact on the social economy and stability. Therefore, network flow anomaly detection has become a key research object, and network flow classification is an important method of that [11]. Flow classification refers to the association of network flow to the applications that generate it. It plays an extremely important role in network management and network security [2]. Especially in network security, flow classification is actually the primary work of anomaly detection and other measures to detect the malicious use of network resources. At present, the mainstream flow classification methods include four categories: port-based methods, DPI-based methods, statistic-based methods, behavior-based methods. The first two methods are rule-based hard coding methods. It is characterized by matching and classification according to fixed rules made manually. The latter two methods are classical machine learning methods. Its feature is to fit the model from the historical data according to the manually selected features for classification [3, 26]. The behavior-based method overcomes the problems of encrypted flow identification and high unpacking cost that cannot be solved by the rule-based method. However, how to choose the appropriate characteristics is a difficult task. Therefore, aiming at the feature dependence problem of malicious traffic classification methods based on traditional machine learning, this paper proposes a malicious traffic classification method based on representation learning. Methods The expression of traffic direction was analyzed, and the expression features were extracted by deep learning method, which had certain novelty in the feature extraction and characterization of network traffic classification. The research is mainly divided into four parts. The first part is to summarize and analyze the research results of domestic and foreign scholars on RL technology and flow anomaly detection technology; The second part is to establish the framework of the RL model and the extensibility test, and introduce the way of data preprocessing; The third part is to validate and analyze the method of RL and the extensibility test; The fourth part is to summarize the research, analyze the deficiencies in the research, and propose the future research direction. The research aims to further improve the accuracy and efficiency of malicious flow classification, and provide strong guarantee for property protection and social stability.

2. Related Works

The current RL algorithm can extract features from the original data, thus avoiding the traditional feature-based artificial selection. Especially in speech recognition and image classification, it has a good application prospect. Sch ö lkopf and others reviewed the basic knowledge of causal reasoning. It was also associated with important open issues in machine learning to analyze the impact of causality on contemporary machine learning. They found that most causal studies were based on a specific causal factor. Therefore, in the research of artificial intelligence and causality, there was a core issue that was the RL of causality. On this basis, the causes and results of machine learning was described, and the main research directions of the intersection of the two disciplines was given [20]. Ericsson and other researchers introduced four main methods and the latest technologies, and explained how to use self-monitoring in various data formats. They also discussed practical issues such as workflow, representation portability, and computing costs. Finally, researchers have studied some important issues in this field. This has laid a solid foundation for future research [6]. Fang X and other scholars proposed a new geometrically enhanced molecular characterization learning method (GEM). The algorithm adopted a special geometric structure to construct. Several special geometry-level adaptive learning strategies were used to learn geometry knowledge. They compared GEM with various baselines under different benchmarks. The results showed that this method was much better than other baselines [7]. Zhou researchers proposed a specific association model to characterize the possible multiple source relationships. First, the different features generated by each encoder were used to estimate the parameters independent of the mode. Then the individual's representation was transformed into possible multi-source correlation features. Finally, the attention mechanism was used to integrate cross-pattern related representation into shared representation, highlighting the most critical segmentation feature. Through the analysis of the BraTS datasets, it is found that the model had a good processing effect. When one or more modes were lost, they still had a robust effect [25].

The research on network security maintenance is of great significance in society. Yang et al. studied a new malicious data flow detection method based on deep learning. The model can automatically extract malicious data from encrypted networks, and has the characteristics of autonomous learning and intelligent adaptability. The model could effectively overcome the problems of sample size and distribution imbalance. Through verification experiments, this method could distinguish normal and unconventional flow, with an accuracy of 99.94%. Experiments showed that this method could effectively improve malicious flow detection in encrypted networks [24]. Arivudambi and other scholars proposed a new flow analysis scheme using relevant methods. They verified it by collecting real-time flow data for a week. The experimental results denoted that the proposed method was superior to the current flow analysis technology. And they showed the excellent flow classification performance of this method, effectively solving the network attack problem of malicious flow [4]. Sharma and other scholars proposed a new network anomaly detection method based on features. On this basis, the optimal feature selection method was used to classify DNS services. In the second stage, an improved HoltWinter method was used to predict the normal behavior in the future. Finally, it needed to determine the location of abnormal data. The experiment indicated that the prediction accuracy of their proposed method has been greatly improved [21]. Nakashima M team proposed an integration technology based on greedy search. This technology solved the termination problem in feature elimination and reduced the number of feature points. The method was tested by two sets of datasets. The results illustrated that the integrated recognition method could achieve comparable performance with traditional selection technology in the case of a small number of features [16].

To sum up, RL has the function of reducing task complexity and saving feature workload. Network flow is characterized by complexity and high difficulty in feature processing. Therefore, the research applied RL to flow detection technology and proposed a flow anomaly detection method based on RL. Its purpose is to help malicious flow be more accurately identified and improve the stability of the network environment. The specific characteristics of related work are shown in Table 1.

Table 1

Comparison table of specific features

Method	Advantage	Shortcoming
GEM [7]	The special geometric structure can capture the intrinsic geometric features of the data, which is suitable for the analysis of complex data structures. The geometric adaptive learning strategy is helpful to improve the adaptive ability and learning efficiency of the model to the data features.	High complexity. It has limited applicability and generality to other types of data
Malicious data stream detection for deep learning [24]	Automatic feature extraction. High accuracy.	High computing resource requirements. Model transparency is low
Specific association models deal with multiple source relationships [25]	Efficient integration of multi-source data. Reinforce key features	The model complexity is high. Strong data dependence
Integration technology based on greedy search [16]	High feature selection efficiency. Strong adaptability	Overfitting is easy to occur. The search efficiency is low
Classification of malicious traffic based on representation learning	Direct use of raw data to reduce feature engineering requirements; High classification performance and accuracy; Adaptability and scalability; Reduce the calculation burden	/

3. Construction of Malicious Flow Classification Model Based on RL

RL aims to extract meaningful and distinguishable features from raw data. When studying the processing of network flow data as images, it is actually a transformation or mapping of the data to reveal its internal structure and patterns in new forms. The connection between the malicious flow classification model based on RL and the expression form of network flow is mainly reflected in how to effectively extract, express, and utilize key information and patterns in flow data to achieve efficient, accurate, and robust network anomaly detection.

3.1. Expressions of Network Flow

Using the flow classification method of machine learning, it is necessary to segment the continuous flow according to a certain granularity, and then obtain multiple discrete cells. Each packet in these discrete units is divided into multiple layers through OSI or TCP/IP [1]. There are many different ways to segment network flow, including TCP connection form, host, service, session and flow. In the same original flow data, it is segmented in different forms. The representation of the datasets is also quite different. Currently, the form of flow and session is widely used in flow classification. Flow refers to all packets of source IP, destination IP, source port, destination port and transport layer protocol. The five tuple that make up the flow are called the same quintuple. The “five tuple” is used in computer networks, especially in IP networks and transport layer protocols, to determine a specific data flow. The source IP address is the IP address of the sender of the data stream; The source port number is the port number of the sender of the data flow, which is usually associated with a specific process or service; The target IP address is the IP address of the recipient of the data flow; The target port number is the port number of the receiver of the data flow, which is usually associated with a specific process or service; Protocol is a field indicating the protocol used by the data flow. Quintuples are commonly used in scenarios such as network monitoring, security, and routing decisions to identify or distinguish different data flows. Sessions refer to all packets composed of two-way flows. Two-way flow means that the source and address in a five-tuple can be interchanged. The original flow can be expressed by Formula (1) [15].

$$P = \{p^1, \cdots, p^{|p|}\}. \tag{1}$$

Formula (1) defines the set of all packets in network traffic and is the starting point for network traffic data analysis. In formula (1), P represents the set of all packages. P^i represents the package. The expression of P^i is shown in Formula (2).

$$P^i = (x^i, b^i, t^i). \tag{2}$$

The quintuple information of Formula (2) is the key attribute used to distinguish different flows in the network flow. In Formula (2), each package consists of three elements. There are five-tuple information x^i, packet length b^i in bytes, and packet transmission time t^i. The flow divides the set into several subsets according to the five-tuple information, as shown in Formula (3).

$$P = \{p^1 = (x^1, b^1, t^1), \cdots, p^n = (x^n, b^n, t^n)\}. \tag{3}$$

Formula (3) divides the packet set into subsets according to the quintuple information, and each subset represents a stream. This is a key step in network traffic analysis, dividing continuous network traffic data into meaningful units for analysis. The packets in each subset are arranged according to the length of time, and the arrangement expression is shown in Formula (4) [12].

$$t^1 < t^2 < \cdots < t^n \cdot \tag{4}$$

Formula (4) indicates that within each subset, packets are arranged according to length of time. This ensures that the packet order within the stream is preserved and is crucial for analyzing the behavior and characteristics of the stream. An expression can be obtained by combining Formulae (3)-(4), which is called a flow, as shown in Formula (5).

$$f = (x, b, d, t). \tag{5}$$

In Formula (5), f represents flow. d_t indicates the duration of the package. The whole raw flow is expressed in the form of flow as shown in Formula (6).

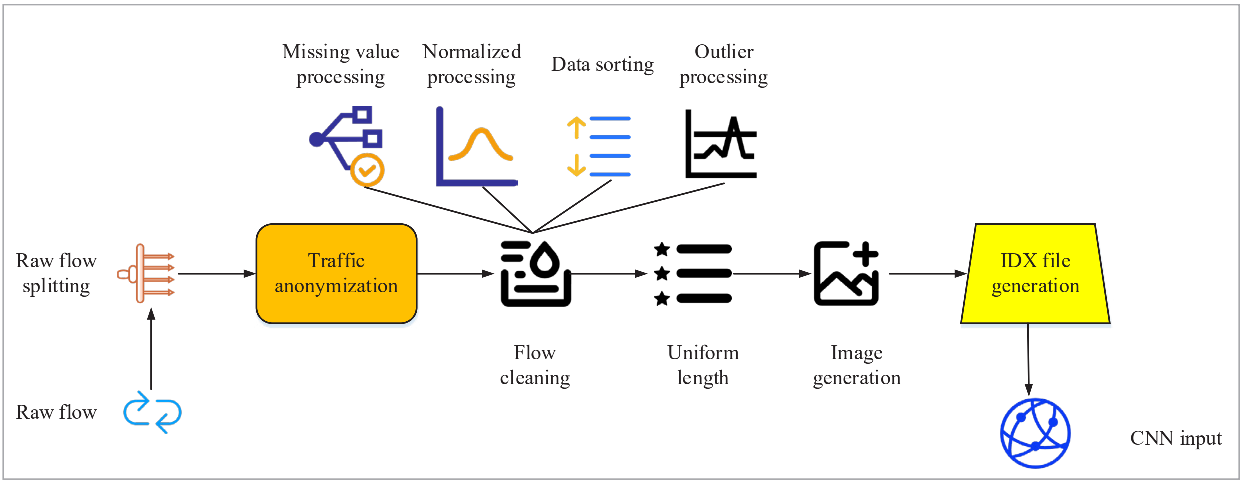
$$F = \{f^1, \cdots f^n\}. \tag{6}$$

Formulae (5)-(6) provide representations of the flow and the entire original flow, emphasizing the importance of time persistence when defining a flow. Sessions are further defined based on the concept of flow, which is especially important for traffic analysis of two-way communications, given that source and destination are interchangeable. The form of the session is the same as the expression of the flow. The difference between the two is whether the source and address in the five-tuple can be interchanged. The form of conversation is interchangeable, so the form of conversation is also called two-way flow. When using streams or sessions, it should be noted that different streams or sessions have different degrees. However, the requirements of the deep learning model on the size of data input must be the same. To solve this problem, the model only selects the initial bytes of the stream or session. This method is adopted because the front end of the flow or session is the data packet that establishes the connection. Such data packets can better reflect the flow characteristics. The packets behind the stream or session are mainly data, which cannot well show the characteristics of flow. These traditional machine learning algorithms use the same methods or ideas when detecting malicious flow. The research only needs to select the first few hundred bytes of the session or stream. This is more portable and simpler than the traditional machine learning algorithm [8].

From the analysis of protocol layer transmission, the application layer is mainly used to reflect the characteristics of flow, which is in the seventh layer of OSI model. It sets SMTP protocol to represent mail flow and HTTP to represent browser flow. Based on the above assumptions, only the application layer is selected as the representation form of flow. However, in other protocol layers, sometimes there is information reflecting the characteristics of flow. For example, the port information of the transport layer can identify most standard interface applications. In the transport layer, different tag bits can also reflect the characteristics of SYN attack and RST attack. There are two main ways to choose the protocol layer. The first way is to use all protocol level data and express it with All. The second method only uses application layer data and is represented by L7. It is worth noting that there are IP addresses and MAC addresses unique to each flow in the "All" data. This information may affect the classification feature extraction to a certain extent. To solve these factors, the unique information of these flow data needs to be processed randomly. The above operation is generally called flow cleaning, which is a part of data preprocessing. Data processing refers to the processing steps from raw flow to data input. After reference, the data processing process is shown in Figure 1 [17].

In Figure 1, the data input format is through different combinations of flow granularity and packet level. The representation of flow can be divided into four forms: flow and "All" combination, flow and L7 combination, conversation and "All" combination, conversation and L7 combination. When the format of

Figure 1
Workflow of Network Flow Data Processing



input data is pcap, the output format is the combination of stream and “All” or the combination of session and “All”. “pcap” is a common network packet capture file format. Through network protocol analyzer tools such as Wireshark, network packet information recorded in pcap files can be read and analyzed. When the input data format is bin, the output format is the combination of stream and L7 or the combination of session and L7. The method of image generation is to convert network traffic data into image form, which involves taking the characteristics of each packet or session and encoding them into a two-dimensional array to obtain a gray image. Figure 1 forms the IDX, DX file format commonly associated with the MNIST database for benchmarking in image processing and machine learning. CNN’s IDX file generator works by processing raw network data and converting it into a format suitable for CNN input. The reason for using IDX is to achieve efficient storage and loading of data, which has obvious effects on large data sets used for training machine learning models.

3.2. Construction of RL Model

The flow picture will be generated during data processing. The scale and size of the processed images are similar to the handwritten recognition datasets. Therefore, the structure of the deep learning model used in the study is a Convolutional Neural Network (CNN). Based on relevant literature experience, the CNN structure constructed in the study is shown in Figure 2 [14].

In Figure 2, the size of the read window of the CNN structure is 28 * 28 * 1, which represents the pixel value of the flow picture. These pixel values need to be normalized and mapped to the range of 0 to 1. In the first convolution layer, 5 * 5 convolution verification data is used for convolution. There are 32 convolutional channels in total, generating 32 28 * 28. The maximum value of 2 * 2 is processed at P1 level, and 32 14 * 14 feature maps are obtained. In the second convolution layer, the number of convolution cores is also 5 * 5. However, a total of 64 channels generates 64 14 * 14 characteristic maps. At the P2 level, the maximum value of 2 * 2 is performed, and 64 characteristic diagrams of 7 * 7 are obtained. Then there are two full connection steps. The full connection converts the data size into 1024 and 10 in sequence. Finally, all possible output values are realized through softmax. To reduce over-fitting, Dropout is used in front of the output layer. Assuming that the flow bytes in a session or stream are represented by k as a vector, the expression of a session or flow is shown in Formula (7) [9].

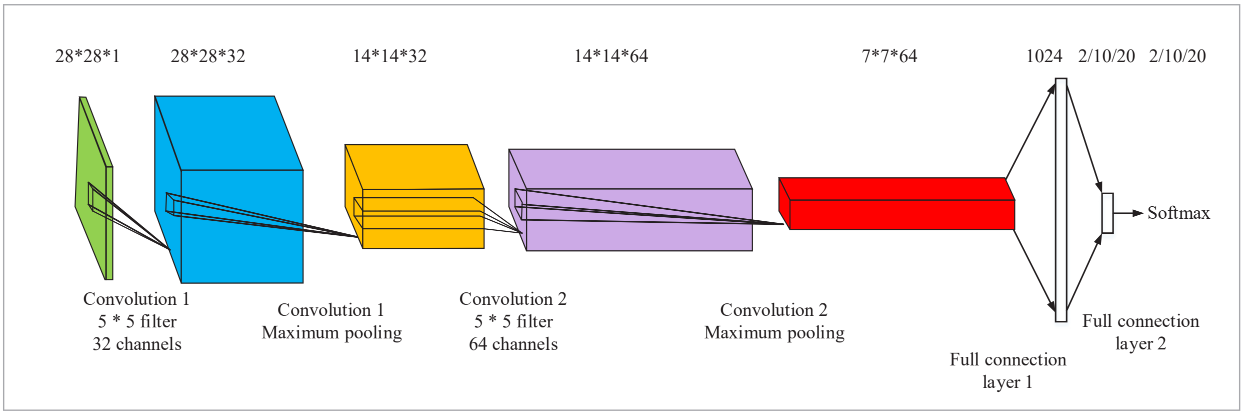
$$x_{1n} = x_1 \oplus x_2 \oplus \cdots \oplus x_n . \tag{7}$$

Formula (7) shows how bytes of a session or stream are represented as vectors, a critical step in feeding network traffic data into a machine learning model. In Formula (7), n represents the length of the session or flow. $\oplus$ represents a join operator. A convolution operation includes a filter or convolution core. Then the filter operates on a group of flow bytes and finally outputs a new feature. The characteristic expression is shown in Formula (8) [10].

$$c_i = R(w \cdot x_{i+h-1} + b) . \tag{8}$$

Figure 2

Structure Diagram of CNN RL Model



In Formula (8), h represents the filter window width; w represents convolution kernel; b stands for offset term, The bias term is a learnable parameter used in machine learning and neural networks to give the model more adjustment space and help it better fit the data. The bias term itself is learned and adjusted during network training through optimization algorithms such as backpropagation and gradient descent. It is not calculated through a fixed formula, but rather a parameter that is iteratively updated as the training process progresses; R represents ReLU nonlinear function. The characteristic mapping of flow bytes is shown in Formula (9) [19].

$$c = [c_1, c_2, \cdots c_{n-h+1}]. \tag{9}$$

Formulae (8)-(9) define the convolution operation, which uses the filter to operate on the flow bytes to output new features. Formula (9) applies the ReLU nonlinear function to the result of the convolution operation to produce the feature mapping. These steps are the basis for CNN's processing of image and image-like data to extract meaningful features from the raw data. The maximum pooling operation is performed on the feature map to obtain the maximum characteristic value. The proposed method needs to be verified for scalability, so two scenarios are set. The two scenarios include three classification forms, including 2 classifier, 10 classifier and 20 classifier. The experiments of the two scenarios are shown in Figures 3-4 [22].

In Figure 3, the focus of scenario A is to test the performance of the model under class 2 and class 10 classifiers. This setup mimics the most common need in real-world applications - identifying malicious data on the network. Not only is this at the heart of most current intrusion detection system (IDS) research, but it also fits the need to initially classify traffic as malicious or normal. In this scenario, by mixing 20 different data streams, the classification effect of two different data streams is compared and analyzed. Scenario A consists of two steps. The first step is a Class 2 classifier, which involves simply dividing traffic into malicious traffic and normal traffic. This step tests the accuracy and efficiency of the model on basic classification tasks; The second step is the 10-class classifier, in which the model is tasked with further subdividing the malicious traffic into 10 different categories. This step tests the model's performance when dealing with more complex classification tasks, including its ability to identify specific malicious behavior [18].

In Figure 4, scenario B is designed to test the model's performance in 20 classifiers. This applies to situations where multiple data streams need to be classified at once with high accuracy, and the challenge is that the model must be able to accurately distinguish between multiple categories of traffic, including various types of malicious traffic and normal traffic. In this setup, the model needs to distinguish between 20 different types of traffic, including 10 malicious traffic and 10 normal traffic. This step is designed to evaluate the accuracy and robustness of the model in classifying multiple complex traffic flows at once. Therefore, datasets are generally feature data that have been manually selected, which do not meet the use requirements in the RL method. In an open datasets, even if the original flow data can be provided, there are too

Figure 3

Extensibility Research Structure of CNN RL Model in Scenario A

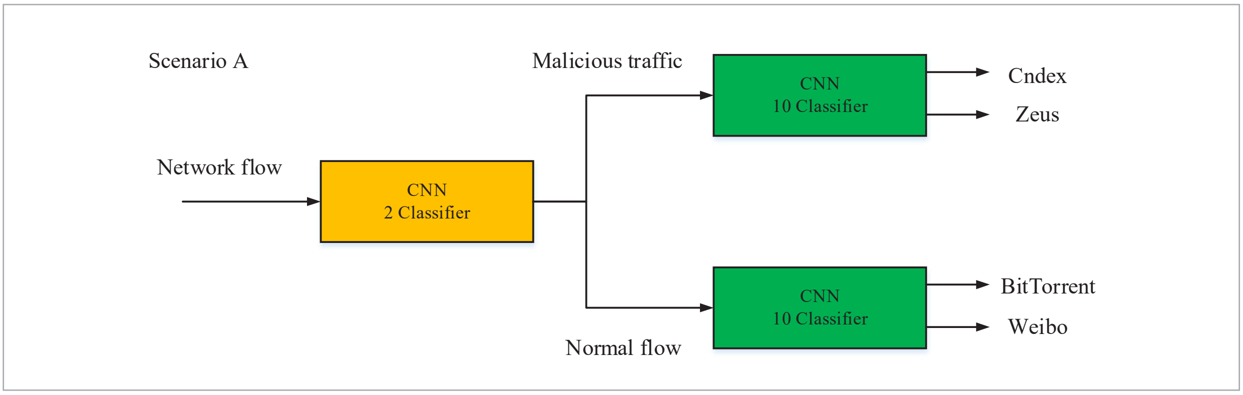
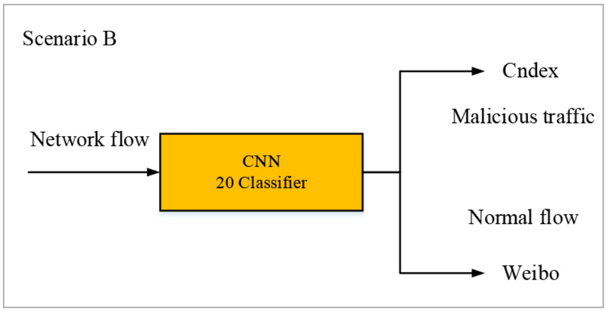


Figure 4

Extensibility Research Structure of CNN RL Model in Scenario B



few data including both normal and malicious flow. To solve the above problems, the study adopts the USTC-TFC2016 datasets [23]. This data has 10 kinds of malicious flow and 10 kinds of normal flow. The samples of malicious flow are collected in the real environment. When some files with large size are used, the research adopts interception processing, and merges the smaller applications. Normal flow is collected by professional simulation equipment. To reflect the diversity of normal flow types, 10 kinds of normal flow cover 8 kinds of common networks. In this study, Accuracy (A), Precision (P) and F1 Score(F1) were used as indicators of the overall effect of the method [13, 5]. During CNN recognition training, the loss rate curve will fluctuate greatly. Therefore, Linear Discriminant Analysis (LDA) is introduced into the CNN model to optimize the classifier. Now assume that there is a spatial dimension in which m samples exist and are expressed as $\{x_1, x_2, \cdots, x_m\}$, then the expression of coefficient in LDA is shown in Formula (10).

$$\begin{cases} u_i = \dfrac{\sum\limits_{x \in classi} X}{n_i} \\ u = \dfrac{\sum\limits_{i=1}^{m} x_i}{m} \end{cases}. \tag{10}$$

In Formula (10), u_i represents the mean value of a single type sample; u represents the mean of all samples; n_i represents the dimension of the sample. LDA will find a new mapping surface during the classification process, and there is no intersection in the mapping process. In order to better identify the mean distance between samples and various categories, the concept of middle distance is further introduced in Formula (10). Its expression is shown in Formula (11).

$$\begin{cases} D_1 = (x_i - u_i)^{T+1} \\ D_2 = \sum\limits_{j=1}^{k} (x_i - u_j)^{T+1} \end{cases}. \tag{11}$$

In Formula (11), D_1 represents the mean distance; D_2 represents the sum of mean distances. Bring the Formula (11) into the Softmax function to get the Formula (12).

$$p(y^{(1)} = j \,|\, x^{(i)}; \theta) = (\theta_j^T (x^{(i)} - u_j)^{T+1}) / (\textstyle\sum_{l=1}^{k} \theta_l^T (x^{(i)} - u_l)^{T+1}). \tag{12}$$

In Formula (12), θ represents the model parameters and p represents the probability of occurrence. The cost function in the new algorithm is turned into the Formula (13).

$$J_\theta = -\frac{[\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{k} 1\{y^{(1)} = j\} (\theta_j^T (x^{(i)} - u_j)^{T+1}) / (\sum_l^k \theta_l^T (x^{(i)} - u_l)^{T+1})]}{m}. \tag{13}$$

By derivation of Formula (13), the descending gradient formula of model parameters can be obtained, and its expression is shown in Equation (14).

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left\{ \frac{D_j}{\sum_{l=1}^{k} D_l \theta_l^T} \left[1\{y^{(1)} = j\} - p(y^{(1)} = j \,|\, x^{(1)}; \theta) \right] \right\} + \lambda \theta_j. \tag{14}$$

If Formula (14) is introduced into the gradient algorithm, the time cost function can be minimized, so as to complete the improvement effect of the classifier.

4. Performance Analysis of Malicious Flow Classification Model Based on RL

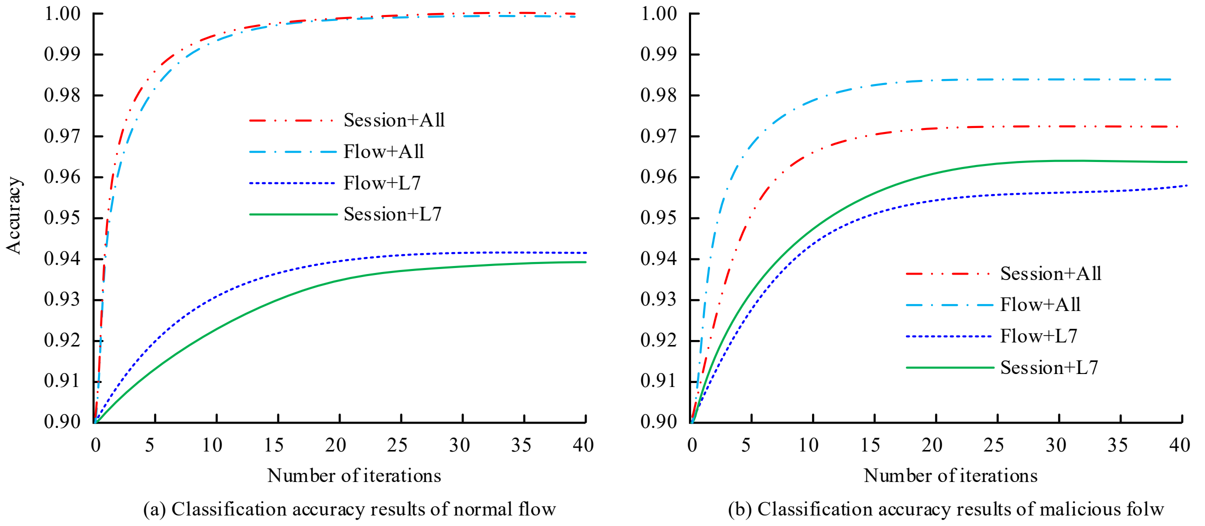
To conduct performance analysis on the constructed model, experiments were conducted using a dataset to explore the optimal representation form of flow. It evaluated the comprehensive performance of the model in identifying malicious flow using metrics such as precision, recall, and F1 Score.

4.1. Analysis of Experimental Results of Network Flow Characterization

After the construction of the flow classification model was completed, the performance of the model was

Figure 5

Classification Accuracy of Network Flow Representation



(a) Classification accuracy results of normal flow

(b) Classification accuracy results of malicious folw

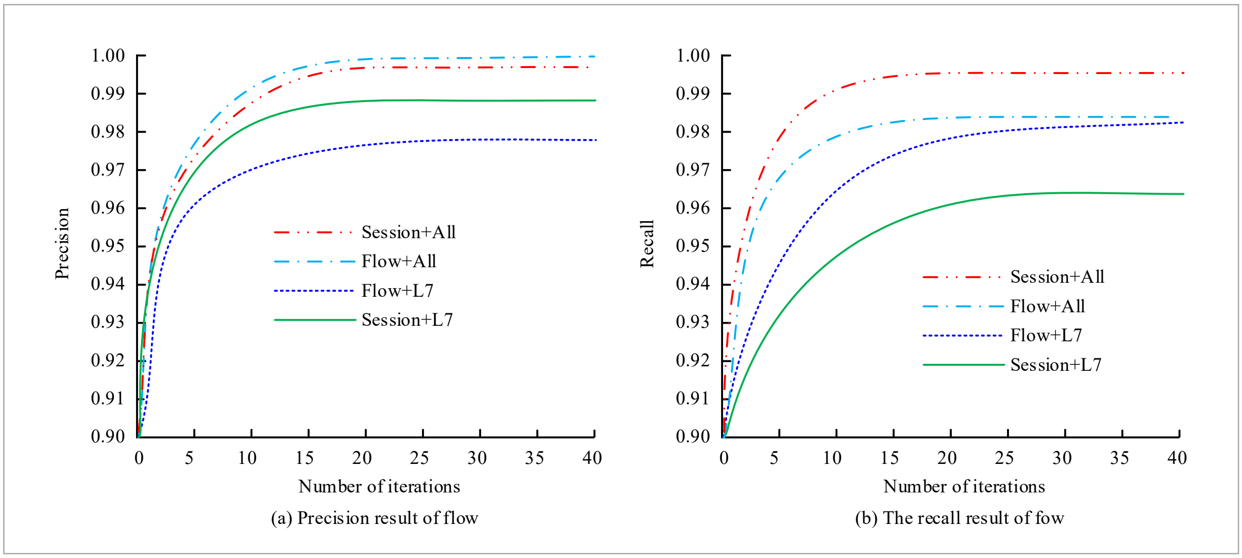
analyzed by experiments. In experimental operation, TensorFlow was used as the software framework. The computer hardware configuration was as follows: the CPU was 16-core XeonE5-2680, and the memory was 16GB. In addition, a GPU was used as an accelerator. GPU accelerator could improve the training speed and the operation efficiency of the model. In the experiment, the data in the datasets was divided into training data and test data according to the ratio of 9:1. The mini-batch size in the model training was set to 50, and the cross-entropy function was used as the loss function. The learning rate was set to 0.001. The number of workouts was set to 40. In the characterization experiment, the study used the USTC-TFC2016 datasets to conduct experiments on four flow characterization forms, and the results are shown in Figure 5.

Figure 5 shows the classification accuracy results of four types of flow representation forms in normal and malicious flow datasets. Figure 5(a) shows the classification accuracy results of normal flow. Among them, the accuracy of flow and All representation in normal flow datasets was 0.998; The accuracy of flow and L7 representation was 0.941; The accuracy of session and All representation was 0.999; The precision of conversation and L7 representation was 0.937. Figure 5(b) shows the classification accuracy results of malicious flow. Among them, the accuracy of stream and All representation in malicious flow datasets was 0.973; The accuracy of flow and L7 representation was 0.959; The accuracy of session and All representation was 0.983; The accuracy of conversation and L7 representation was 0.962. From all layers and application layers, the accuracy of All representation was higher than that of L7 representation in the same datasets; From the analysis of conversation and flow, in the L7 representation of normal flow, there was no obvious difference between the accuracy of conversation and flow, and the accuracy of other forms was higher than that of flow.

Figure 6(a) shows the precision result of the flow. The precision of flow and All forms was 0.999; The precision of flow and L7 form was 0.978; The precision of conversation and All was 0.997; The precision of conversation and L7 was 0.989. Figure 6(b) shows the recall result of the flow. Among them, the recall rate of stream and all forms was 0.983; The recall rate of flow and L7 was 0.964; The recall rate of conversation and All was 0.995; The recall rate between the session and L7 was 0.982. After comprehensive analysis of the above results, except that the recall rate of conversation and all forms was 0.002 lower than that of stream and all forms, the precision and recall rate of all forms were higher than that of L7 forms; The precision and

Figure 6

Precision and recall results of four expressions of random flow



(a) Precision result of flow

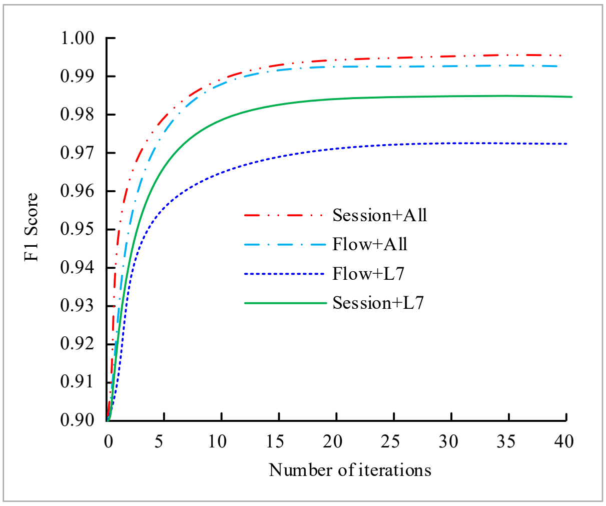
(b) The recall result of fow

recall of conversation form were higher than that of stream form.

Figure 7 shows F1 Score in different expressions of a random flow. In Figure 7, F1 Score of flow and All was 0.993; F1 Score of flow and L7 was 0.972; F1 Score of session and All was 0.994; The F1 Score of session and L7 was 0.984. The experimental results showed that the F1 Score of All form was higher than that of L7 form; The F1 Score of the session form was higher than that of the stream form. An explanation can be given for the above results. The conversation form has two-way flow, so it contains more interactive information. This form can represent more information than separate flows.

Figure 7

F1 Results of Random Flow



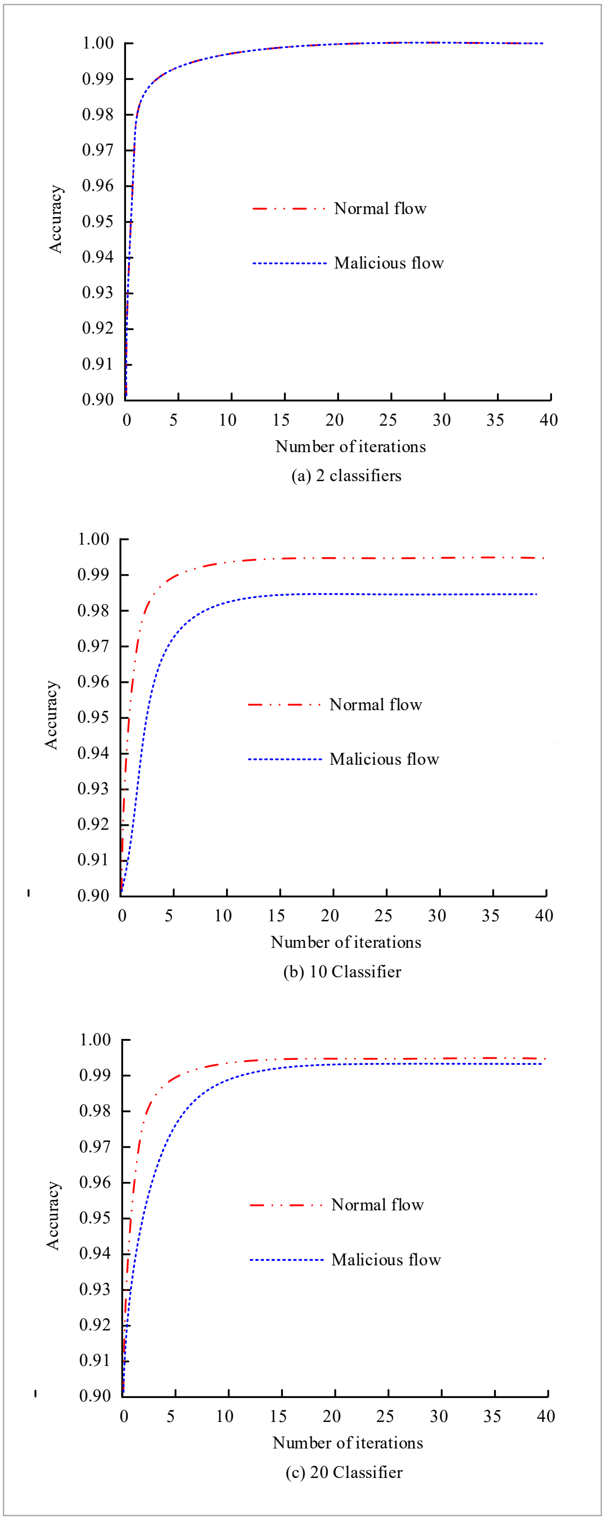
4.2. Extensibility Experiment and Classification Result Analysis of the Model

Through the characterization experiment of flow, it is determined that session and All were the best representation of flow data. In the extensibility verification part, the data representation of flow was in the form of session and All. The research used this representation method to verify the effect of three classifiers in two different scenarios. Scenario A used two classifiers first, and then used non-conventional classifiers for classification; Scenario B was directly classified by twenty classifiers, and the classification effect is shown in Figure 8.

Figure 8 shows the classification average accuracy of the three classifiers in the classification process, where the red curve represents the change in recognition accuracy of normal flow and the blue curve represents the change in recognition accuracy of malicious flow. In Figure 8(a), the classification accuracy of both normal traffic and malicious traffic of the binary classifier converges to 100% as the number of

Figure 8

Average Classification Accuracy of Three Classifiers



iterations increases. In Figure 8(b), the normal traffic classification accuracy of the ten classifier begins to converge at the fifth iteration and finally stabilizes at 99.93%. The classification accuracy of malicious traffic starts to converge at the fifth iteration and finally stabilizes at 98.61%. In Figure 8(c), the normal flow classification of the twenty-class device begins to converge at the 7th iteration and finally stabilizes at an accuracy of 99.41%. The classification accuracy of malicious traffic begins to converge at the 10th iteration and finally stabilizes at 99.17%. The experimental results show that the three classifiers have high accuracy.

Table 2 shows the classification effect of the non-constant classifier for each type of flow. In the data in Table 1, only Virut and Neris had slightly lower indicators. Among them, the precision rate of Virut was only 88.92%, that of Neris was 97.04%, and that of other flows was above 99%; The recall rate of Virut was 96.4%, that of Neris was 91.07%, and the precision rate of other flow was above 99%; The F1 value of Virut was 92.56%, that of Neris was 93.95%, and the F1 value of other flows was above 99%.

Table 3 shows the classification effect of twenty-classifiers for each flow. In the data in Table 3, only Virut and Neris had slightly lower indicators. Among them, the accuracy rate of Virut was only 90.63%, that of Neris was 96.28%, and that of other flows was above 98%; The recall rate of Virut was 95.52%, that of Neris was 92.83%, and the precision rate of other flows was above 98%; The F1 value of Virut was 93.04%, that of Neris was 94.58%, and the F1 value of other flows was above 98%. The value of Viru and Neris flow was low, which might be related to the specific application characteristics. On the whole, the three classifiers met the requirements of practical applications, indicating the feasibility of the proposed representation method.

Figure 9 shows the classification effect of normal data traffic. In Figure 9, four types of traffic, Wow, Weibo, FTP and SMB, have obvious clustering effect, and also have outstanding detection effect in the experiment. Although Facetimet and BitTrt have certain clustering effects, they are not easily distinguishable from other types of clustering traffic due to their close distance. The experimental results also show that the detection effects of Facetime and BitTrt are relatively poor.

Table 2

Classification Effect of the Ten-classifier for Each Flow

Category	P	R	F1	Category	P	R	F1
Zeus	99.94%	99.93%	99.93%	Wow	99.90%	99.98%	99.92%
Virut	88.92%	96.43%	92.56%	Weibo	99.97%	99.96%	99.90%
Tinba	99.81%	99.90%	99.86%	SMB	99.94%	99.94%	99.98%
Shifu	99.88%	99.77%	99.79%	Skype	99.91%	99.92%	99.95%
Nsis-ay	99.65%	99.24%	99.42%	Outlook	99.58%	99.89%	99.62%
Neris	97.04%	91.07%	93.95%	MySQL	99.95%	99.96%	99.98%
Miuref	99.92%	99.91%	99.95%	Gmail	99.82%	99.63%	99.73%
Htbot	99.79%	99.78%	99.78%	FTP	99.98%	99.90%	99.98%
Geodo	99.96%	99.95%	99.91%	Facetime	99.96%	99.97%	99.92%
Cridex	99.93%	99.92%	99.94%	BitTrt	99.94%	99.94%	99.95%

Table 3

Classification Effect of Twenty-classifiers for Each flow

Category	P	R	F1	Category	P	R	F1
Zeus	99.93%	99.97%	99.97%	Wow	99.97%	99.84%	99.81%
Virut	90.63%	95.52%	93.04%	Weibo	99.98%	99.95%	99.92%
Tinba	99.96%	99.93%	99.92%	SMB	99.90%	99.93%	99.97%
Shifu	99.89%	99.84%	99.80%	Skype	99.72%	99.95%	99.85%
Nsis-ay	99.72%	99.07%	99.36%	Outlook	99.16%	98.02%	98.63%
Neris	96.28%	92.83%	94.58%	MySQL	99.91%	99.91%	99.97%
Miuref	99.95%	99.92%	99.94%	Gmail	98.37%	99.28%	98.75%
Htbot	99.78%	99.98%	99.88%	FTP	99.93%	99.95%	99.92%
Geodo	99.91%	99.89%	99.89%	Facetime	99.90%	99.96%	99.94%
Cridex	99.94%	99.94%	99.91%	BitTrt	99.98%	99.92%	99.92%

Figure 9

Classification Effect of Normal Data Flow

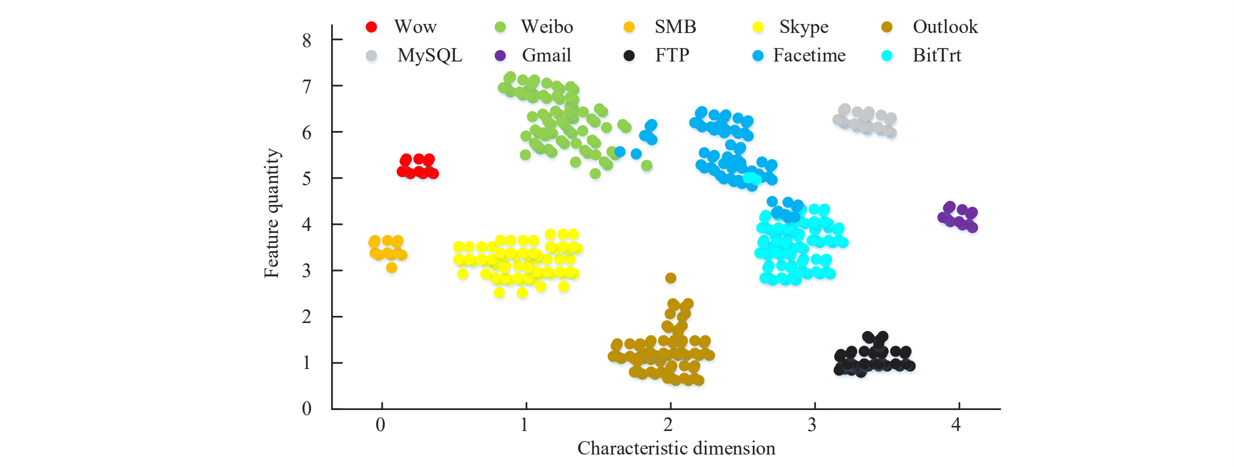


Figure 10

Classification Effect of Malicious Data flow

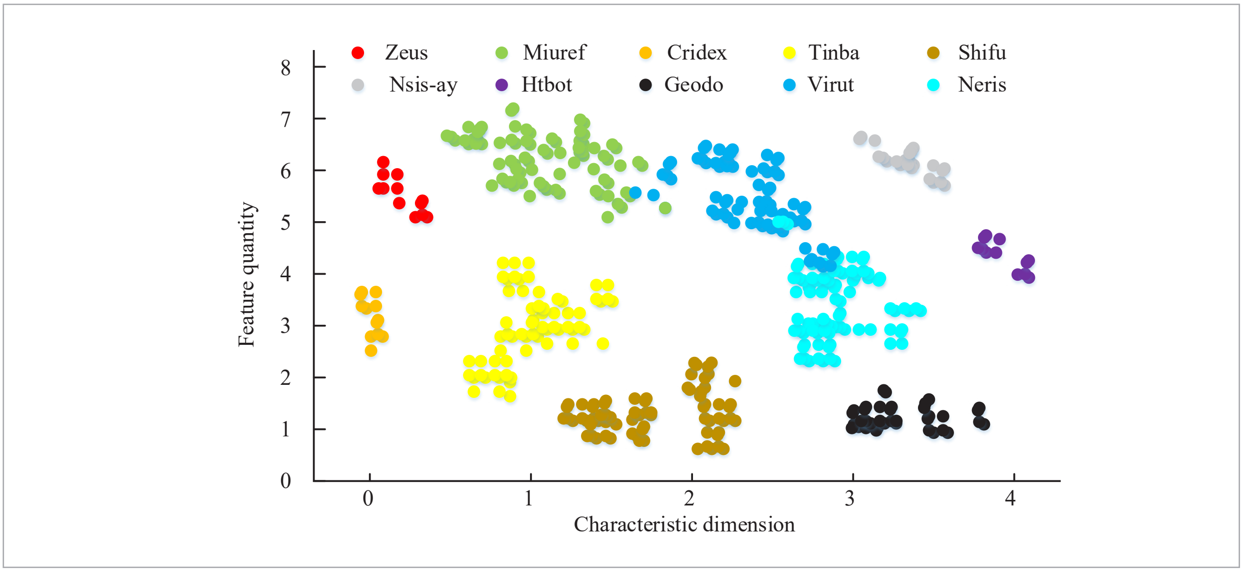


Figure 10 shows the classification effect of malicious data traffic. In Figure 10, except Shifu and Htbot, the other traffic has obvious clustering effect, indicating that the malicious traffic is well distinguished. Shifu and Htbot also have some clustering effect, but there is a small overlap with other traffic, indicating that some traffic is difficult to distinguish. In order to further verify the performance of the proposed method, a set of data containing malicious traffic was used for experiments, and the algorithm was evaluated by ten indicators. The evaluation indexes include true case rate (TPR), false positive case rate (FPR), true negative case rate (TNR), false negative case rate (FNR), Accuracy, Precision, Recall, F1 Score, AUC value and error rate. There are 1000 samples of traffic data, of which 213 are malicious traffic data. Table 4 shows the specific results.

In Table 4, the accuracy rate of the representation learning method reaches 0.9830, which is much higher than the GEM algorithm's 0.9560 and Greedy

Table 4

Algorithm evaluation and comparison results

Index	GEM	Greedy search	Representation Learning
TPR	771	778	785
FPR	16	9	2
TNR	185	192	198
FNR	29	22	15
Accuracy	0.9560	0.9700	0.9830
Precision	0.9797	0.9886	0.9974
Recall	0.9638	0.9725	0.9813
F1 Score	0.9717	0.9805	0.9893
AUC	0.9652	0.9834	0.9932
Error	0.0440	0.0300	0.0170

search technology's 0.9700, indicating that it is more accurate and reliable in identifying malicious traffic. Similarly, the accuracy and recall performance of the representation learning method also indicate its advantages in reducing false positives (FPR of only 2) and catching true positives (TPR of 785), which is particularly important for malicious traffic detection, as too high a false positive rate can lead to unnecessary alerts, while missing true malicious traffic can be a security concern. In addition, the F1 score and AUC values representing the learning method reached 0.9893 and 0.9932, respectively, further demonstrating its ability to balance accuracy and recall and maintain excellent performance under different thresholds. The Error rate (Error) is only 0.0170, which is much lower than the other two methods, which means that there will be fewer error classifications when using the presentation learning method. The results show that the representation learning method has significant performance advantages on several key indicators of comprehensive evaluation of malicious traffic detection capability. This approach significantly improves the accuracy and efficiency of malicious traffic detection by making more effective use of data features and improving the generalization ability of the model, making it a preferred algorithm in similar scenarios.

5. Conclusion

The research explored the anomaly detection of network flow, and proposed a method to classify malicious flow using RL method. This method characterized different flow in the form of flow or session. It generated a visual graph of flow through CNN model and classified it. Through experimental verification, the representation of conversation and All had a classification accuracy of 0.999; The precision rate was 0.997; The recall rate was 0.995; F1 Score was 0.994, which was the best characterization method of flow data. Different classifiers had high classification accuracy for the data of this representation method. The two-classifier accuracy reached 100%; The ten-classifier accuracy for normal flow and malicious flow was 99.93% and 98.61%, respectively; The accuracy of twenty-classifiers reached 99.43%. The experimental results showed that the proposed method had good malicious flow identification performance and could be applied in practice. However, there were still deficiencies in the study. In flow detection, the detection effect of malicious flow was more uniform, while the detection effect of normal flow was unstable. This may be because the sources of normal and abnormal flow are different. Therefore, the follow-up experiment aims to improve the flow source, eliminate the above problems, and further improve the stability and accuracy of malicious flow detection.

References

1. Abbasi, M., Shahraki, A., Taherkordi, A. Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey. Computer Communications, 2021, 170, 19-41. https://doi.org/10.1016/j.comcom.2021.01.021
2. Akbari, I., Salahuddin, M. A., Ven, L., Limam, N., Boutaba, R., Mathieu, B., Moteau, S., Tuffin, S. A Look Behind the Curtain: Traffic Classification in an Increasingly Encrypted Web. Performance Evaluation Review, 2021, 49(1), 23-24. https://doi.org/10.1145/3543516.3453921
3. Alzahrani, R. J., Alzahrani, A. Survey of Traffic Classification Solution in IoT Networks. International Journal of Computer Applications, 2021, 183(9), 37-45. https://doi.org/10.5120/ijca2021921392
4. Arivudainambi, D., Kumar, K., Satapathy, S. C. Correlation-Based Malicious Traffic Analysis System. International Journal of Knowledge-Based and Intelligent Engineering Systems, 2021, 25(2), 195-200. https://doi.org/10.3233/KES-210064
5. Chen, Y., Yang, J., Chen, B., Du, S. Counting Varying Density Crowds Through Density Guided Adaptive Selection CNN and Transformer Estimation. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 33(3), 1055-1068. https://doi.org/10.1109/TCSVT.2022.3208714
6. Ericsson, L., Gouk, H., Loy, C. C., Hospedales, T. M. Self-Supervised Representation Learning: Introduction, Advances, and Challenges. IEEE Signal Processing Magazine, 2022, 39(3), 42-62. https://doi.org/10.1109/MSP.2021.3134634
7. Fang, X., Liu, L., Lei, J., He, D., Zhang, S., Zhou, J., Wu, H., Wang, H. Geometry-Enhanced Molecular Representation Learning for Property Prediction. Nature Machine Intelligence, 2022, 4(2), 127-134. https://doi.org/10.1038/s42256-021-00438-4
8. Ghafoori, M. S., Soltani, J. Designing a Robust Cyber-Attack Detection and Identification Algorithm for DC Mi-

crogrids Based on Kalman Filter with Unknown Input Observer. IET Generation, Transmission & Distribution, 2022, 16(16), 3230-3244. https://doi.org/10.1049/gtd2.12517

9. Gopal, S. B., Poongodi, C., Nanthiya, D., Kirubakaran, T., Kulavishnusaravanan, B., Logeshwar, D. Autoencoder-Based Architecture for Identification and Mitigating Phishing URL Attack in IoT Using DNN. Journal of The Institution of Engineers (India), Series B. Electrical Engineering, Electronics and Telecommunication Engineering, Computer Engineering, 2023, 104(6),1227-1240. https://doi.org/10.1007/s40031-023-00934-8

10. Guo, J., Jia, R., Su, R., Song, Y., Jing, F. DoS Attack Detection in Identification of FIR Systems with Binary-Valued Observations. Asian Journal of Control: Affiliated with ACPA, the Asian Control Professors Association, 2023, 25(4), 2469-2481. https://doi.org/10.1002/asjc.3005

11. Hubballi, N., Khandait, P. KeyClass: Efficient Keyword Matching for Network Traffic Classification. Computer Communications, 2022, 185, 79-91. https://doi.org/10.1016/j.comcom.2021.12.021

12. Li, D., Gebraeel, N., Paynabar, K., Meliopoulos, A. An Online Approach to Covert Attack Detection and Identification in Power Systems. IEEE Transactions on Power Systems: A Publication of the Power Engineering Society, 2023, 38(1), 267-277. https://doi.org/10.1109/TPWRS.2022.3167024

13. Li, J., Zhang, D., Meng, B., Li, Y., Luo, L. FIMF Score-CAM: Fast Score-CAM Based on Local Multi-Feature Integration for Visual Interpretation of CNNs. IET Image Processing, 2023, 17(3), 761-772. https://doi.org/10.1049/ipr2.12670

14. Li, P. H., Xu, J., Xu, Z. Y., Chen, S., Niu, B. W., Yin, J., Sun, X. F., Lan, H. L., Chen, L. L. Automatic Botnet Attack Identification Based on Machine Learning. Computers, Materials & Continua, 2022, 73(2), 3847-3860. https://doi.org/10.32604/cmc.2022.029969

15. Munther, A., Abdulrazzaq, A., Abualhaj, M. M., Almukhaini. Reduce Memory Consumption for Internet Traffic Classification. International Journal of Networking and Virtual Organisations, 2021, 24(2), 144-160. https://doi.org/10.1504/IJNVO.2021.114730

16. Nakashima, M., Sim, A., Kim, Y., et al. Automated Feature Selection for Anomaly Detection in Network Traffic Data. ACM Transactions on Management Information Systems (TMIS), 2021, 12(3), 1-28. https://doi.org/10.1145/3446636

17. Patil, S., Bhavikatti, A. M. Performance Analysis and Design of Automatic Attack Identification for Ad-Hoc Wireless Channel and Improvement in Coverage. International Journal of Computational Intelligence Theory and Practice, 2022, 17(2), 119-129.

18. Qiu, W., Li, C., Tang, Q., Sun, K., Liu, Y., Yao, W. Attack Detection for Spoofed Synchrophasor Measurements Using Segmentation Network. CSEE Journal of Power and Energy Systems, 2022, 8(5), 1327-1337.

19. Rani, S. V. J., Charan, S., Prakash, S., Parekh, N., Ioannou, I., Christophorou, C., Vassiliou, V., Pitsillides, A., Nagaradjane, P. Detection of DDoS Attacks in D2D Communications Using Machine Learning Approach. Computer Communications, 2023, 198(2), 32-51. https://doi.org/10.1016/j.comcom.2022.11.013

20. Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., Bengio, Y. Toward Causal Representation Learning. Proceedings of the IEEE, 2021, 109(5), 612-634. https://doi.org/10.1109/JPROC.2021.3058954

21. Sharma, R., Guleria, A., Singla, R. K. Flow-Based Profile Generation and Network Traffic Detection for DNS Anomalies Using Optimized Entropy-Based Features Selection and Modified Holt Winter's Method. International Journal of Security and Networks, 2021, 16(4), 244-257. https://doi.org/10.1504/IJSN.2021.119380

22. Sun, Y., Hou, L., Lv, Z., Peng, D. Informer-Based Intrusion Detection Method for Network Attack of Integrated Energy System. IEEE Journal of Radio Frequency Identification, 2022, 6(1), 748-752. https://doi.org/10.1109/JRFID.2022.3215599

23. Wang, B., Su, Y., Zhang, M., Nie, J. A Deep Hierarchical Network for Packet-Level Malicious Traffic Detection. IEEE Access, 2020, 8, 201728-201740. https://doi.org/10.1109/ACCESS.2020.3035967

24. Yang, J., Liang, G., Li, B., Wen, G., Gao, T. A Deep Learning and Reinforcement Learning Based System for Encrypted Network Malicious Traffic Detection. Electronics Letters, 2021, 57(9), 363-365. https://doi.org/10.1049/ell2.12125

25. Zhou, T., Canu, S., Vera, P., Ruan, S. Latent Correlation Representation Learning for Brain Tumor Segmentation with Missing MRI Modalities. IEEE Transactions on Image Processing, 2021, 30, 4263-4274. https://doi.org/10.1109/TIP.2021.3070752

26. Zhu, M. Y., Chen, Z., Chen, K. F., Lv, N., Zhong, Y. Attention-Based Federated Incremental Learning for Traffic Classification in the Internet of Things. Computer Communications, 2022, 185, 168-175. https://doi.org/10.1016/j.comcom.2022.01.006